

# Data Collection

## Big Data & Predictive Analytics

Mulia Sulistiyono, M.Kom

# Sumber Data



# Jenis data

## Structured vs Unstructured

Big Data memiliki tiga karakteristik, salah satunya adalah **variety (variasi)**. Variety disini maksudnya adalah data pada Big Data memiliki berbagai macam jenis data.

Dimulai dari jenis data yang terstruktur hingga seiring berkembangnya teknologi Big Data, jenis data mulai tidak terstruktur. Lalu apa bedanya dari masing-masing jenis data tersebut?

# Jenis data

## 1. Data terstruktur

Jenis data structured (data tradisional) dapat diproses, disimpan, dan diambil dalam format tetap.

Jenis data ini disimpan dalam bentuk tabel, baris dan kolom yang normalnya disimpan dalam excel atau spreadsheet, dimana informasi pada data sangat terorganisir dan dapat dengan mudah diakses dari database dengan algoritma mesin pencari sederhana.

Contoh data terstruktur adalah, data sensor, data penjualan pada suatu perusahaan, data karyawan dalam database perusahaan dengan detail yang terstruktur seperti detail data diri karyawan, posisi pekerjaan, gaji, dan lainnya ditampilkan secara terorganisir.

# Jenis data

## 2. Data semi-terstruktur

Jenis data semi-structured merupakan jenis data yang dimasukkan ke dalam sebuah tabel, tetapi skemanya tidak sama dengan tabel biasa yang hanya terdiri dari baris dan kolom.

Data semi-terstruktur mengandung format data terstruktur dan tidak terstruktur. Walaupun belum diklasifikasi oleh repository tertentu (database), namun mengandung informasi yang penting.

Contohnya adalah data dalam bentuk file csv, file xml, dan file json.



# Jenis data

## 3. Data tidak terstruktur

Jenis data unstructured adalah data dengan **bentuk yang tidak dikenal**, harus disimpan dengan format khusus karena tidak memiliki struktur yang spesifik seperti jenis data structured.

Raw data dari jenis data ini hanya **dapat menghasilkan nilai setelah diproses dan dianalisa**. Menyimpan data jenis ini pun memiliki kerumitan seperti memerlukan penggunaan sistem penyimpanan yang memadai, seperti database NoSQL (MongoDB dan CouchDB).

Contoh jenis data tidak terstruktur seperti data teks, berformat foto atau gambar, video, atau suara. Selain itu, bisa juga dalam bentuk keluhan pelanggan, kontrak, ataupun email internal. Contoh dari data jenis ini dapat ditemukan dalam social media, seperti komentar, likes, followers, dan data click pada setiap aktivitas di akun media sosial.

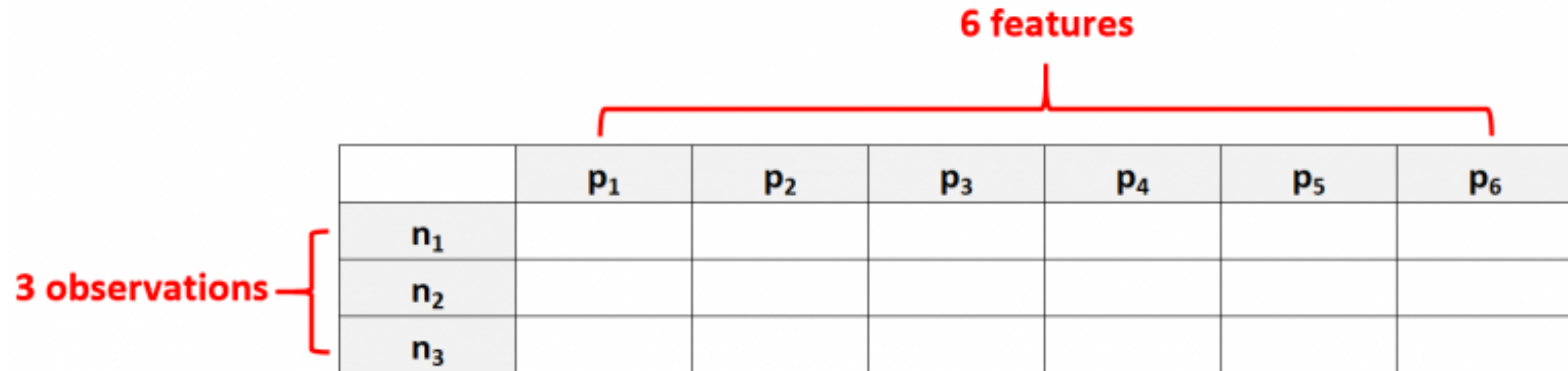
# Steps to collect Big Data

Suatu data mentah tanpa adanya tahapan pembersihan dan pengujian sebelumnya tidak akan berarti (not-valuable). Data yang menghasilkan value harus **terstruktur dengan baik dan bersih**.

1. Gather data
2. Store data
3. Clean up data
4. Reorganize data
5. Verify data

# High Dimensional data

Data berdimensi tinggi mengacu pada dataset dimana **jumlah fitur  $p$  lebih besar** dari **jumlah pengamatan/observation  $N$** , sering ditulis sebagai  **$p \gg N$** .



The diagram illustrates a data matrix with 3 rows and 6 columns. The columns are labeled  $p_1$  through  $p_6$ , and the rows are labeled  $n_1$  through  $n_3$ . A red bracket above the columns is labeled "6 features", and a red bracket to the left of the rows is labeled "3 observations".

	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$
$n_1$						
$n_2$						
$n_3$						



One common mistake people make is assuming that  
“high dimensional data”

A dataset could have 10,000 features, but if it has  
100,000 observations then **it's not high  
dimensional.**

# Why is High Dimensional data a Problem?

- Ketika jumlah fitur dalam dataset melebihi jumlah pengamatan, kita tidak akan pernah memiliki jawaban deterministik.
- Dengan kata lain, menjadi tidak mungkin untuk menemukan model yang dapat menggambarkan hubungan antara variabel independen dan variabel dependen.
- Hal ini terjadi karena kita tidak memiliki cukup observasi untuk melatih model tersebut.

# How to Handle High Dimensional Data

Terdapat 2 cara untuk mengatasi high dimensional data:

## 1. **Menggunakan sedikit features**

Beberapa cara untuk memutuskan fitur mana yg akan dihapus

- Hapus features dengan missing value yang tinggi
- Hapus features dengan varians rendah
- Hapus features yang memiliki korelasi rendah dengan variable dependen.

## 2. **Menggunakan metode regularization:** Principian Component Analysis (PCA), Ridge Regression atau Lasso Regression.

# Web penyedia Dataset

1. Satu data Indonesia,  
<https://data.go.id/home>
2. Kaggle datasets,  
<https://www.kaggle.com/search?q=indonesia+in:datasets>
3. UCI Machine Learning Repository,  
<https://archive.ics.uci.edu/ml/index.php>

# Membaca Dataset

1. Menggunakan **library pandas**.
2. Download dataset dari penyedia data.
3. Jalankan dan cek hasilnya pada notebook (Jupyter atau Colab)



# Code

```
In [2]: import pandas as pd

df = pd.read_csv('diabetes.csv')
df.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

# Code

```
In [5]: df.shape
```

```
Out[5]: (768, 9)
```

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 768 entries, 0 to 767
```

```
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	Pregnancies	768 non-null	int64
1	Glucose	768 non-null	int64
2	BloodPressure	768 non-null	int64
3	SkinThickness	768 non-null	int64
4	Insulin	768 non-null	int64
5	BMI	768 non-null	float64
6	DiabetesPedigreeFunction	768 non-null	float64
7	Age	768 non-null	int64
8	Outcome	768 non-null	int64

```
dtypes: float64(2), int64(7)
```

```
memory usage: 54.1 KB
```

# Referensi

1. Yuk, Kenali 3 Jenis Data Pada Big Data yang Wajib Kamu Tahu!, DQLab, <https://www.dqlab.id/kenali-tiga-jenis-data-pada-big-data>
2. 5 Steps to collect Big Data, Octoparse, <https://www.octoparse.com/blog/5-steps-to-collect-big-data#>
3. What is High Dimensional Data? (Definition & Examples), Statology, <https://www.statology.org/high-dimensional-data/>